

# Quantum Algorithm for the Computation of the Reactant Conversion Rate in Homogeneous Turbulence

Guanglei Xu<sup>a</sup>, Andrew J. Daley<sup>b</sup>, Peyman Givi<sup>c</sup>, and Rolando D. Somma<sup>d</sup>

<sup>a</sup>Department of Physics and SUPA, University of Strathclyde, Glasgow G4 0NG, United Kingdom, email: glxu.leo@gmail.com; <sup>b</sup>Department of Physics and SUPA, University of Strathclyde, Glasgow G4 0NG, United Kingdom, email: andrew.daley@strath.ac.uk; <sup>c</sup>Mechanical Engineering and Chemical & Petroleum Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA, email: peg10@pitt.edu; <sup>d</sup>Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA, email: somma@lanl.gov.

## ARTICLE HISTORY

Compiled July 31, 2019

## ABSTRACT

The rapid developments that have occurred in quantum computing platforms over the past few years raise important questions about the potential for applications of this new type of technology to fluid dynamics and combustion problems, and the timescales on which such applications might be possible. As a concrete example, here a quantum algorithm is developed and employed for predicting the rate of reactant conversion in the binary reaction of  $F + rO \rightarrow (1 + r) \text{ Product}$  in non-premixed homogeneous turbulence. These relations are obtained by means of a transported probability density function equation. The quantum algorithm is developed to solve this equation and is shown to yield the rate of the reactants' conversion much more efficiently than current classical methods, achieving a quadratic quantum speedup, in line with expectations for speedups arising from quantum metrology techniques more broadly. This provides an important example of a quantum algorithm with a real engineering application, which can build a connection to present work in turbulent combustion modelling and form the basis for further development of quantum computing platforms and their applications to fluid dynamics.

## 1. Introduction

Estimation of the conversion rate of chemical reactants has been the subject of broad investigations in combustion literature for over seven decades [1]. In non-premixed systems, there are two factors by which this rate is influenced: (1) the speed at which the reactants are brought into the reaction

zone and (2) the rate at which they are converted to products through chemical reactions. The rate of reactant conversion is characterized by the Damköhler number ( $Da$ ). As  $Da \rightarrow \infty$ , the limiting rate of conversion is achieved. In most analysis of turbulent reacting flows, the statistical mean value of this rate is of both theoretical and practical interest [2–5]. The probability density function (PDF) methodology has been particularly effective for estimating this value [6–11], and the transport equation for the PDF is typically simulated numerically via classical Monte-Carlo (MC) methods [12].

This work explores to what extent the solutions to problems of this type could be sped up on a quantum computing architecture. Quantum computing is undergoing rapid development across a range of platforms [13–20]. In this potentially transformative paradigm, the underlying physical properties of microscopic objects are used to encode and process information. This makes it possible to take advantage of quantum physics to perform certain classes of computations using very different methods to traditional classical computing. It can lead to large relative speedups, usually quantified by how the number of elementary operations required to solve a problem scales with the problem size (see, for example, Ref. [21]).

Inherently, quantum computing derives its possible speedup over classical computing due to some properties of quantum physics that have no classical analogue. One such property is superposition: the underlying units of information - so-called quantum bits, or *qubits*, can be placed in a superposition state that encodes information in its amplitudes. Another property is quantum entanglement, where quantum states with non-classical correlations exist and the state of each qubit cannot be described independently from the state of the other qubits. Quantum interference is also a key property as it is the reason behind the preparation of quantum states in superposition that are mostly supported in those states that encode information about the solution to a problem. Quantum algorithms that present important quantum speedups, such as Shor’s factoring algorithm [22], exploit the properties discussed above.

Remarkably, quantum computing hardware has already evolved to a stage where it is important to identify key short and long-term applications, and to begin to optimize hardware towards the most promising applications. The challenge is then to design algorithms that could run on such hardware. The development of quantum algorithms, while hard, is then exceptionally important.

In a recent article [23], the present authors developed a quantum algorithm for accurately predicting properties of turbulent fluid flows. These results are another contribution to the significant and recent work in developing quantum algorithms for related search and sampling problems [24–32]. In

particular, recent results in *quantum metrology* have shown that with a quantum algorithm, it is possible to achieve a quadratic complexity improvement over current classical algorithms [28, 31, 33]. In more detail, classical probabilistic methods for parameter estimation output an estimate within precision of order  $1/\sqrt{N_r}$ , where  $N_r$  is the number of computational steps performed (*e.g.*, the number of samples or repetitions). In contrast, there exist quantum computing methods that provide the estimate within precision of order  $1/N_r$ , where  $N_r$  now refers to the number of basic quantum operations and should be then compared to the number of classical computational steps. Fixing a particular level of precision would then lead to the conclusion that quantum computers would require quadratically fewer resources than traditional classical computers for parameter estimation. These quantum metrology algorithms were conceptually adapted to compute quantities that otherwise would be computed using classical MC sampling methods [26].

In Ref. [23], the algorithm based on Ref. [28] was used to simulate turbulent mixing. This was demonstrated by consideration of the classical scalar mixing modelled via the coalescence/dispersion (C/D) closure [34–36]. In the present work, the objective is to demonstrate that the methodology has the potentials for applications in a wider range of classical problems. The applicability of the method of Ref. [23] to the more complex case of chemically reactive turbulent flows is here analyzed.

It is important to remark that by generalizing and extending the application areas of quantum algorithms, as done in this paper, further and needed connections between scientific communities investigating turbulent reacting flows and developing quantum algorithms will result.

## 2. Formulation

In what follows, the error scaling of classical MC methods and of the quantum algorithm when applied to the specific problem of reactive flow is analyzed. The general formulation of this problem, which is specialized for the relevant examples in §3, is now introduced.

The problem regards the transport of two initially segregated reactants  $F(\mathbf{x}, t)$  and  $O(\mathbf{x}, t)$ , where  $\mathbf{x}-t$  denote the (homogeneous) space-time. In particular, an idealized irreversible binary reaction of the type  $F + rO \rightarrow (1+r) \text{ Product}$  with initially segregated reactants is considered. The turbulence field is assumed to be statistically homogeneous, and all of the chemical species are assumed to have identical and constant thermodynamic properties. The quantity  $P(\Psi, t)$  denotes the joint PDF of the two reactants ( $\Psi : \psi_1 \equiv F, \psi_2 \equiv O$ ) and, at the initial time ( $t = 0$ ), the two species are assumed

to be totally segregated:

$$P(\Psi, 0) = W_F \delta(\psi_1 - F_0) \delta(\psi_2) + W_O \delta(\psi_1) \delta(\psi_2 - O_0), \quad W_F + W_O = 1. \quad (1)$$

Here,  $F_0$  and  $O_0$  are the respective initial mass fractions of the two species, and  $W_F$  and  $W_O$  are the respective weight factors of the two reactants. For modelling of the reactant conversion rate, the family of coalescence/dispersion (C/D) mixing models is considered by the evolution equation for the joint PDF [35, 36]:

$$\begin{aligned} \frac{\partial P(\Psi, t)}{\partial t} = & -2\beta\omega P(\Psi, t) \\ & + 2\beta\omega \int d\Psi' \int d\Psi'' P(\Psi', t) P(\Psi'', t) \\ & \times \int_0^1 d\alpha A(\alpha) \delta[\Psi - (1-\alpha)\Psi' - \frac{1}{2}\alpha(\Psi' + \Psi'')] \\ & - \frac{\partial}{\partial \psi_\gamma} [S_\gamma(\Psi) P(\Psi, t)] . \end{aligned} \quad (2)$$

$S(\Psi)$  denotes the chemical source term with its rate measured by the Damköhler number, and summation is implied over the index  $\gamma$ .  $A(\alpha)$  is the PDF of the random variable  $\alpha$ ,  $0 \leq \alpha \leq 1$ . The value of  $\alpha$  determines the conditions of mixing. For the example here, Curl's model is used [37], in which  $A(\alpha) = \delta(\alpha - 1)$ . Other models can be readily used as well. The parameter  $\omega$  is the mixing frequency and determines the rate of variance decay. The parameter  $\beta$  depends on  $A(\alpha)$  as follows:

$$\beta = \frac{1}{a_1 - \frac{1}{2}a_2}, \quad a_m = \int_0^1 d\alpha \alpha^m A(\alpha) . \quad (3)$$

Through integration of Eq. (2) all of the pertinent single-point statistics of the reacting field can be determined. The most important of these statistics are the ensemble mean values of the reactants' concentrations, denoted by  $\langle \dots \rangle$ . Furthermore,  $r = 1$  with  $F_0 = O_0 = 1$  are assumed, and the species are introduced in stoichiometric proportion:  $W_F = W_O = \frac{1}{2}$ . With this, the Shvab-Zeldovich variable [2, 3] (or the mixture fraction [38]) is defined:

$$\mathcal{J}(\mathbf{x}, t) = F(\mathbf{x}, t) - O(\mathbf{x}, t) . \quad (4)$$

For reactive flows with non-equilibrium chemistry (finite Damköhler (Da) numbers), full numerical solution of Eq. (2) is required. In the limit of infinitely fast chemistry ( $\text{Da} \rightarrow \infty$ ), the statistics

of the two reactants are related to those of the Shvab-Zeldovich variable  $\mathcal{J}$ . Through integration of Eq. (2) all of the pertinent single-point statistics of the reacting field can be determined. Of significant importance is the rate of reactant conversion measured by:

$$\mathcal{Z}(t) = 1 - \frac{\langle F \rangle(t)}{\langle F \rangle(0)}. \quad (5)$$

### 3. Results

Taking a system that is described by Curl’s model as an example, the quantum algorithm provides the estimation of the rate of reactant conversion in Eq. (5),  $\mathcal{Z}(t)$ , within certain precision. The important variable to quantify the complexity of the quantum algorithm is  $N_r$  that, as mentioned, refers to the number of repetitions, or times a certain quantum state has to be prepared. This quantum state encodes information about the probabilities of an analogue classical MC method that also provides an estimate of  $\mathcal{Z}(t)$ . Then, to analyze the advantages of the quantum algorithm,  $N_r$  should be compared with the number of repetitions of the MC method to provide the estimate within the same precision and confidence levels.

MC methods simulate a PDF derived from the original one, Eq. (2), after a number of approximations. These approximations are mainly based on discretizations of continuous parameters, including the time evolution. The actual PDF that MC methods simulate result in the mean rate of reactant conversion  $\mathcal{Z}_{\text{MC}}(t)$ . As the size of the discretization parameters become smaller, it is expected that  $\mathcal{Z}_{\text{MC}}(t)$  converges, in some sense, to  $\mathcal{Z}(t)$ . Here, the approximation error  $|\mathcal{Z}_{\text{MC}}(t) - \mathcal{Z}(t)|$  is *not* analyzed, since both the quantum algorithm and classical MC will provide an estimate of  $\mathcal{Z}_{\text{MC}}(t)$  within the given precision. The main result is that the quantum algorithm requires quadratically less resources than classical MC to obtain an estimate of  $\mathcal{Z}_{\text{MC}}(t)$  within the same precision.

#### 3.1. Simulations via classical MC methods

To demonstrate the performance of classical MC as well as to provide a basis for comparison with the quantum algorithm, MC simulations are conducted of Eq. (2). The limit of infinitely fast chemistry is considered first. In this case, the marginal form of Eq. (2) is considered with  $\Psi \equiv \psi$  for the Shvab-Zeldovich variable. For this mixing controlled condition, the source term for the Shvab-Zeldovich variable is  $S = 0$ . For finite rate chemistry calculations, the joint PDF of two reactants is

considered ( $\Psi : \psi_1 \equiv F, \psi_2 \equiv O$ ). The mixing operations for  $\psi_1$  and  $\psi_2$  are identical to that for  $\psi$ , but the chemistry terms are accounted for with inclusion of  $S(\Psi, t)$ . For clarity of the presentation below, the subscripts for  $\psi$  are dropped unless they are needed. In all calculations, the mixing frequency is (arbitrarily) set to  $\omega = 1$ .

To simulate Eq. (2) via classical MC methods, a certain number of computation elements or ‘particles’,  $N_p$ , is chosen. For each element, a random variable  $\psi^k(i, t_j)$  is associated. Here,  $k = 1, 2, \dots, N_r$  refers to the MC run, i.e., there are  $N_r$  executions of the MC method;  $i = 1, 2, \dots, N_p$  refers to a specific particle; and  $t_j = j \cdot \Delta t$  is the physical time at the  $j$ -th step in any run, which has been discretized using time steps of size  $\Delta t$ . In MC, the joint PDF of all particles is  $\mathcal{Q}(\psi^k(1, t_j), \psi^k(2, t_j), \dots, \psi^k(N_p, t_j))$  and simulates  $P(\Psi, t)$ . The initial distribution  $\mathcal{Q}(\psi^k(1, 0), \psi^k(2, 0), \dots, \psi^k(N_p, 0))$  shall represent  $P(\Psi, 0)$ , and is independent of  $k$ . After the  $k$ -th MC run, a random vector  $(\psi^k(1, t_q), \dots, \psi^k(N_p, t_q))$  is generated. Each such vector can be used to estimate quantities such as the reactant conversion rate, or to generate an estimate of the PDF  $\mathcal{Q}(\psi^k(1, t_q), \psi^k(2, t_q), \dots, \psi^k(N_p, t_q))$ , for instance, by building a histogram. The precision of the estimates will depend on  $N_r$ ,  $N_p$ , and  $\Delta t$ .

Simulations are conducted for  $0 \leq t \leq 3$  with  $\Delta t = 0.1$  and  $N_p = 10^3$ . The initial state in each classical MC run is determined according to the initial state of the original PDF. For the Shvab-Zeldovich variable when  $Da \rightarrow \infty$ ,  $P(\psi, 0) = \frac{1}{2}\delta(\psi - 1) + \frac{1}{2}\delta(\psi + 1)$ . This initial PDF is simulated in MC by setting  $\psi^k(i, 0) = -1$ , for all particles labelled by  $i$  such that  $1 \leq i \leq N_p/2$ , and  $\psi^k(i, 0) = +1$  otherwise. The superscript  $k$  refers to the  $k$ -th MC run and satisfies  $1 \leq k \leq N_r$ . The MC method provides an estimate of  $\mathcal{Z}_{MC}(t)$  for different values of  $t$ . The precision of the estimate improves as the number of MC runs increases. In particular, after a total of  $N_r$  runs, the estimation of the limiting rate of reactant conversion is calculated as

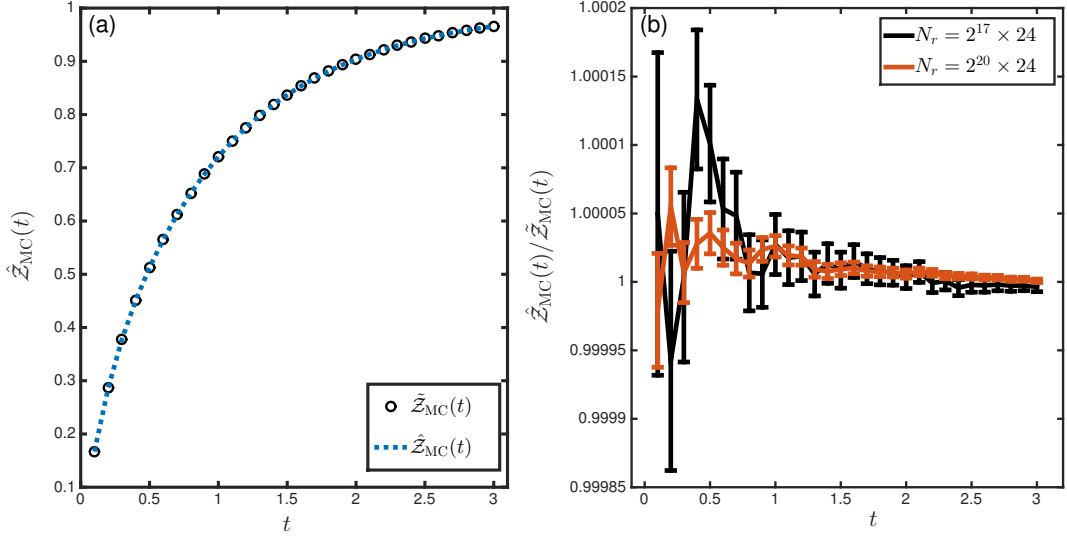
$$\hat{\mathcal{Z}}_{MC}(t) := \frac{1}{N_r} \sum_{k=1}^{N_r} \hat{\mathcal{Z}}_{MC}^k(t) . \quad (6)$$

For  $Da \rightarrow \infty$ , in each MC run a random vector  $(\psi^k(1, t), \dots, \psi^k(N_p, t))$  is generated to calculate the estimation:

$$\hat{\mathcal{Z}}_{MC}^k(t) := 1 - \frac{1}{\langle F \rangle (0)} \frac{1}{N_p} \sum_{i=1}^{N_p} \left( H(\psi^k(i, t)) \cdot \psi^k(i, t) \right) , \quad (7)$$

where  $H(\psi)$  is the Heaviside step function  $H(\psi) = 0.5(1 + \text{sgn}(\psi))$ . For finite  $Da$  values, the effects of

chemistry are taken into account by inclusion of the chemical source term on the MC particles ( $\Psi^k : \psi_1^k, \psi_2^k$ ) [6]. The initial condition for the joint PDF is  $P(\Psi, 0) = \frac{1}{2}\delta(\psi_1)\delta(\psi_2 - 1) + \frac{1}{2}\delta(\psi_1 - 1)\delta(\psi_2)$ . This initial PDF is simulated in MC by setting  $\psi_1^k(i, 0) = 0, \psi_2^k(i, 0) = +1$ , for all particles labelled by  $i$  such that  $1 \leq i \leq N_p/2$ , and  $\psi_1^k(i, 0) = +1, \psi_2^k(i, 0) = 0$ , otherwise. The estimation of the reactant conversion rate or any of the moments are directly calculated over the  $\Psi$  field.



**Figure 1.** Estimates of the rate of reactant conversion obtained from MC simulations of a reacting flow process using Curl's model. (a) Growth of the estimated rate of reactant conversion as a function of time. The blue dotted line is for  $N_r = 2^{17} \times 24$  and the black circles are for a more accurate estimate using  $N_r = 2^{20} \times 60$ . The estimated quantities are  $\hat{Z}_{MC}(t)$  and  $\tilde{Z}_{MC}(t)$ , respectively. (b) Comparison of two estimated rates of reactant conversion for  $N_r = 2^{17} \times 24$  (black line) and  $N_r = 2^{20} \times 24$  (red line) with  $\tilde{Z}_{MC}(t)$  of (a). The relative errors, determined by the error bars, decrease as  $t$  increases.

The results for infinitely fast chemistry obtained from MC simulations are shown in Fig. 1. The behaviour of  $\hat{Z}_{MC}(t)$  as a function of time for  $N_r = 2^{17} \times 24$  MC runs is shown in Fig. 1 (a). These results are compared with a more precise estimate of  $Z_{MC}(t)$ , denoted by  $\tilde{Z}_{MC}(t)$ , obtained from  $N_r = 2^{20} \times 60$  MC runs. A comparison between  $\hat{Z}_{MC}(t)$  for two different values of  $N_r$ , normalised by  $\tilde{Z}_{MC}(t)$ , is given in Fig. 1 (b). To obtain the error bars of Fig. 1 (b),  $\hat{Z}_{MC}^k(t)$  is first computed according to Eq. (7) for each run  $k = 1, \dots, N_r$ . The estimated standard deviation associated with the average value  $\hat{Z}_{MC}(t)$  [Eq. (6)] is calculated as

$$\hat{\sigma}_{Z_{MC}}(t) = \left[ \frac{\sum_{k=1}^{N_r} (\hat{Z}_{MC}^k(t) - \hat{Z}_{MC}(t))^2}{(N_r - 1)} \right]^{1/2}. \quad (8)$$

To reach a 99.75% confidence level for the estimation of  $\mathcal{Z}_{\text{MC}}(t)$ , the statistical error should be

$$\epsilon_C(t) := 3\hat{\sigma}_{\mathcal{Z}_{\text{MC}}}(t), \quad (9)$$

which is defined as the estimation error of the MC simulation in this setting. The error bars of Fig. 1 (b) denote the regions

$$\left[ \frac{\hat{\mathcal{Z}}_{\text{MC}}(t) - \epsilon_C(t)}{\tilde{\mathcal{Z}}_{\text{MC}}(t)}, \frac{\hat{\mathcal{Z}}_{\text{MC}}(t) + \epsilon_C(t)}{\tilde{\mathcal{Z}}_{\text{MC}}(t)} \right]. \quad (10)$$

The simulation results of Fig. 1 and the estimation error of Eq. (9) result in a scaling of the estimation errors with respect to the number of runs of order  $1/\sqrt{N_r}$ .

### 3.2. Simulations via quantum algorithms

The quantum algorithm developed here is based on quantum phase estimation, similar to that of Ref. [23]. An actual implementation of the quantum algorithm would require the development of the corresponding quantum hardware for a problem of an appropriate size-scale, which does not yet exist. This section analyzes then the quantum resources that would be required for an actual implementation. To this end, classical simulation results of the quantum algorithm are also presented.

To describe the algorithm, a brief explanation of quantum computing is necessary. The elementary unit of quantum information in quantum computing is a qubit. Unlike classical computation, a qubit's state can be in a superposition of the 0 and 1 states. In standard notation, these states are described as  $|0\rangle$  and  $|1\rangle$ . The coefficients in the superposition can be complex and satisfy a normalization condition [39]. A quantum computer is built upon many qubits ( $n$ ) and a generic quantum state is then a superposition of all possible basis states. The number of such basis states grows exponentially with the number of qubits as  $2^n$ . The allowed operations on the state of the quantum computer are unitary transformations, which are typically described as a sequence of simpler one and two-qubit operations. That sequence forms a quantum circuit whose complexity is mainly determined by the number of such simple operations. Each operation on the quantum state can modify all of its amplitudes, providing the opportunity to manipulate large amounts of data.

Quantum algorithms are then formulated in three parts. The first part is the initial state prepara-



tion, the second part is specified by a quantum circuit that acts on the initial state [39, 40], and the third part is the measurement that provides classical information about the problem to be solved. To obtain good statistics, the three parts must be repeated sufficiently many times. The difficulty in programming a quantum computer lies then in finding ways to take advantage of properties of quantum mechanics, including the ability of preparing quantum states in superposition.

The quantum algorithm of Ref. [23] was designed by adapting the tools of quantum metrology (*i.e.*, high-precision quantum measurements) of Ref. [28] to turbulent mixing problems. A similar approach is followed here to solve reacting flow problems. In particular, the same problem as that of §3.1 is considered. The following is a summary of the main steps of the quantum algorithm for reacting flow problems. These steps have to be repeated for each value of  $t$ . A more detailed description can be found in Ref. [23].

- First is the preparation of a many-qubit quantum state, denoted by  $|\psi\rangle$ , which is a linear superposition over states in the computational basis. The amplitudes in this superposition encode the same sampling probabilities as that of the classical MC method for some value of  $t$ . This is the part where the physical processes are simulated, and involves the application of essentially *classical* operations on the quantum state [23]. Such operations can be implemented on a quantum computer via two-qubit gates using standard techniques [39].
- Second is the construction of a unitary operation  $U$ , which has an eigenvalue  $e^{i\theta}$ .  $U$  is designed such that  $\mathcal{Z}_{MC}(t) = \cos(\theta/2)$ .
- Third is the computation of an estimate of the eigenphase  $\theta$ , labelled by  $\hat{\theta}_i$ , using the phase estimation algorithm (PEA).
- The above procedure is repeated  $L$  times, where  $L$  depends on the confidence level of the estimation, and  $i = 1, \dots, L$ . The output of the quantum algorithm is the median value of these  $L$  phase estimations, denoted by  $\hat{\theta}$ .

The desired (estimated) quantity is then obtained by using its connection to the (estimated) eigenphase:  $\hat{\mathcal{Z}}_Q(t) = \cos(\hat{\theta}/2)$ .

The optimal application of the operations in the first step will depend on details of the quantum hardware, and this is not the part of the algorithm where a speedup will arise, as the number of operations is of the order of that of a single classical MC run. The speedup of the quantum algorithm as a whole arises instead from reducing the number of samples required to obtain a particular accuracy for a particular parameter of interest, as shown below.

The definition of  $N_r$  for the quantum algorithm is somewhat different from that in the classical MC method, but as in the classical case represents the number of repetitions and therefore quantifies the computational cost of obtaining a sufficiently accurate estimate. Here,  $N_r$  is the total number that a unitary operation  $V$ , which prepares the initial state as  $V|0\dots 0\rangle = |\psi\rangle$ , is implemented. The state  $|0\dots 0\rangle$  is a many-qubit trivial state where all qubits are initialized in the computational basis state  $|0\rangle$ . If the implementation of each PEA returns an estimate  $\hat{\theta}_i$  within  $m$  bits of precision, and  $M = 2^m$ ,

$$N_r = M \times L . \quad (11)$$

Since  $\mathcal{Z}_{MC}(t) \in [0, 1]$ , the range of  $\theta$  is  $[0, \pi]$ :

$$\theta = 2 \arccos(\mathcal{Z}_{MC}(t)) . \quad (12)$$

Each PEA uses  $m$  ancillary qubits that encode the estimate of  $\theta$  upon measurement. In binary form, this estimate is

$$\hat{\theta}_i = 2\pi[b'_1 \dots b'_m] = \pi(b'_1 + b'_2/2 + \dots + b'_m/2^{m-1}) . \quad (13)$$

$b'_j$  is the output state of the  $j$ -th ancillary qubit in the PEA obtained after measurement. The values of  $b'_j$  can then be either 0 or 1. The probabilities of outcomes of the last bit is given by

$$\Pr(b'_m = 0) = \frac{1}{2} (1 + \cos(M\theta)) , \Pr(b'_m = 1) = 1 - \Pr(b'_m = 0) . \quad (14)$$

This probability is obtained as a result of the implementation of the PEA and is well discussed in Ref. [28]. For the remaining bits  $b'_j$ , the probabilities of obtaining outcomes 0 or 1, are given recursively. In particular, for  $j = m - 1, m - 2, \dots, 1$ , these probabilities are

$$\Pr(b'_j = 0) = \frac{1}{2} (1 + \cos(2^j \theta - \pi[b'_{j+1} \dots b'_m])) , \Pr(b'_j = 1) = 1 - \Pr(b'_j = 0) . \quad (15)$$

The output probabilities of the quantum algorithm depend on  $\theta$ , the quantity to be estimated. It is possible then to simulate the quantum algorithm on a classical computer by assuming knowledge of  $\theta$  and then randomly sampling the bits  $b'_j$  according to the above probabilities. Clearly, if  $\theta$

is assumed to be known a priori, no algorithm is necessary to obtain its estimate. However, the classical simulation of the quantum algorithm under this strong assumption still provides useful information in terms of the quantum resources that will be needed in the general case where  $\theta$  is unknown. As shown below, these resources will depend on the precision and confidence level required for the estimate.

The precision level of the estimation of  $\theta$ ,  $\epsilon_\theta$ , is of order of the least significant bit and then  $\epsilon_\theta \leq 2\pi/M$ . From Eq. (12), the precision in the estimation of  $\mathcal{Z}_{MC}(t)$  using the quantum algorithm, noted as  $\epsilon_Q$ , is

$$\epsilon_Q := \left| \cos((\hat{\theta} + \epsilon_\theta/2)/2) - \cos(\hat{\theta}/2) \right|. \quad (16)$$

When  $\epsilon_\theta \ll 1$ ,  $\epsilon_Q$  is of order  $1/M$  and, from Eq. ((11)), this is also of order  $L/N_r$ . Then, the precision  $\epsilon_Q$  depends on the inverse of  $N_r$ , proving a quadratic quantum speedup with respect to classical MC methods to achieve the same precision level. Increasing the number of repetitions  $L$  only increases the confidence level, but not the precision. After  $L$  repetitions of quantum PEA algorithm, the median value of these estimates is the output of our quantum algorithm ( $\hat{\theta}$ ). The confidence level  $c$ , which is the probability that  $\hat{\theta}$  is within precision  $\epsilon_Q$ , can be bounded [28, 41]:

$$c \geq 1 - \frac{1}{2} (0.8)^L. \quad (17)$$

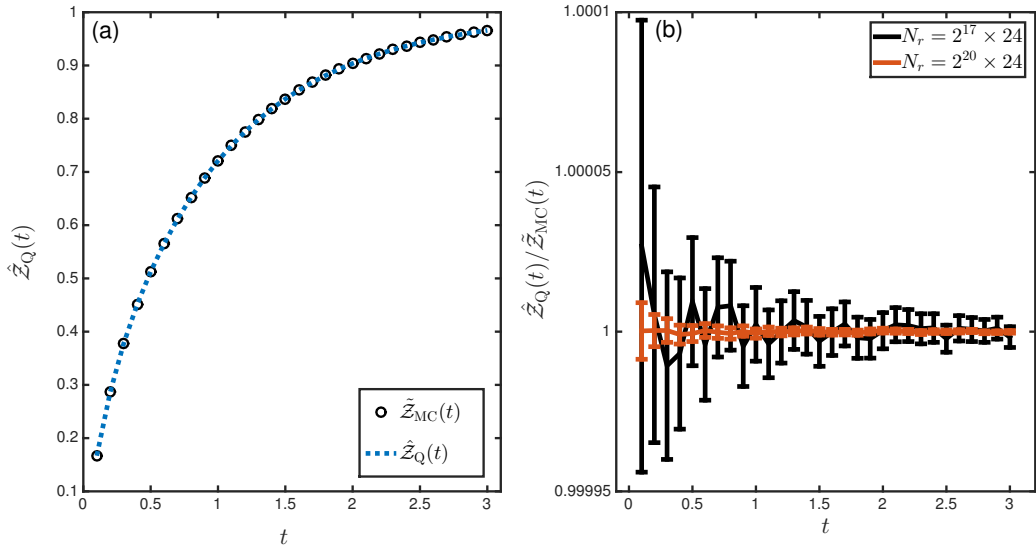
Thus, to reach a desired confidence level  $c$ , the number of repetitions  $L$  is

$$L \geq \frac{\log(2(1-c))}{\log(0.8)}. \quad (18)$$

For a confidence level  $c = 99.75\%$ ,  $L$  is chosen to be 24.

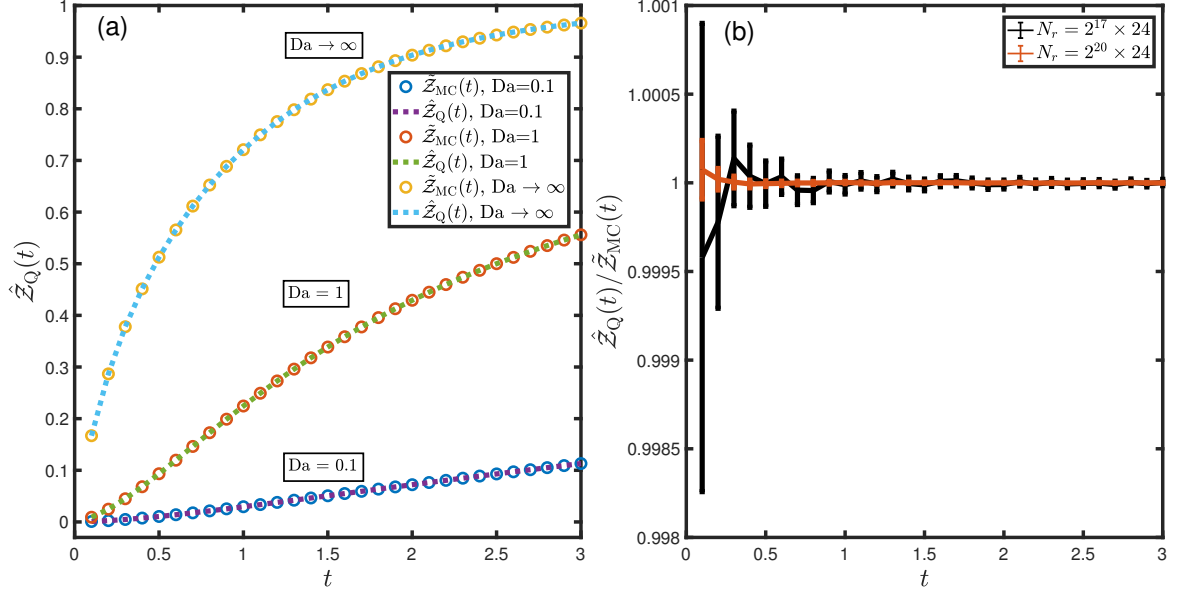
Figure 2 shows classical simulations of the quantum algorithm for estimating  $\mathcal{Z}_{MC}(t)$ , according to the above procedure for sampling the bits  $b'_j$ . The values as a function of time  $t$  for  $0 \leq t \leq 3$  are plotted. The quantum algorithm also takes  $N_p$ ,  $\Delta t$ ,  $\omega$ , and  $\beta$  as input parameters. As in the previous section, these are set to  $N_p = 10^3$ ,  $\Delta t = 0.1$ ,  $\omega = 1$ , and  $\beta = 2$ . The increasing behaviour of  $\mathcal{Z}_{MC}(t)$  as a function of  $t$  is demonstrated in Fig. 2 (a). The results are for  $m = 17$  ancillary qubits used in the PEA, *i.e.*,  $m$  bits of precision in the estimate of the eigenphase  $\hat{\theta}$ . The number of eigenphase estimates is  $L = 24$  and the output is obtained from  $\hat{\mathcal{Z}}_Q(t) = \cos(\hat{\theta}/2)$ . The results of the quantum algorithm are in agreement with the highly accurate results obtained via classical

MC calculations. As in the previous section, these highly-accurate MC calculations were obtained from  $N_r = 2^{20} \times 60$  runs, allowing to obtain estimates  $\tilde{Z}_{MC}(t)$  that are very close to the actual value of  $Z_{MC}(t)$ . The estimated rates of reactant conversion from the quantum algorithm relative to  $\tilde{Z}_{MC}(t)$ , for two values of  $N_r$ , are shown in Fig. 2 (b). In this case, the quantum algorithm was simulated for  $m = 17$  and  $m = 20$  bits of precision in the PEA, and  $L = 24$ . The error bars were obtained according to Eq. ((16)). The relative errors decrease as a function of time because the  $Z(t)$  approaches 1 as  $t$  increases. The extension for finite rate kinetics requires MC simulation of the joint PDF transported equation. The results showing the estimates of the mean rate of reactant conversion obtained from classical simulations of the quantum-algorithm are presented in Fig. 3, where the simulation parameters are the same as those of Fig. 1 except for different Da values.



**Figure 2.** Estimates of the rate of reactant conversion obtained from classical simulations of the quantum-algorithm that would solve a reacting flow process using Curl’s model. The input parameters are the same as those of Fig. 1. (a) Growth of the estimated rate of reactant conversion as a function of time. The blue dotted line is the quantum-algorithm simulation for  $N_r = 2^{17} \times 24$  and the black circles are for the accurate MC estimate  $\tilde{Z}_{MC}(t)$  obtained using  $N_r = 2^{20} \times 60$ . (b) Comparison of two estimated rates of reactant conversion for  $N_r = 2^{17} \times 24$  (black line) and  $N_r = 2^{20} \times 24$  (red line), given by the quantum algorithm, with  $\tilde{Z}_{MC}(t)$ . A different scale than that of Fig.1(b) is used here because of different error ranges. Smaller error values within these estimates are already obtained for values of  $N_r$  where  $m = 17$  and 20.

The parameters used in the simulations of the quantum algorithm are such that they provide a fair comparison with the previous MC simulations. It is observed that, for  $N_r \geq 2^{17} \times 24$ , the quantum algorithm provides more precise estimations of  $Z_{MC}(t)$  than classical MC methods. Figures 4-5 show comparisons of estimation errors from classical MC methods ( $\epsilon_C$ ) and the quantum algorithm ( $\epsilon_Q$ ). These errors are plotted as a function of  $N_r$ . It is observed that  $\epsilon_C$  decreases as  $1/\sqrt{N_r}$  while  $\epsilon_Q$  decreases as  $1/N_r$ , demonstrating a quadratic quantum speedup of the quantum algorithm with respect to MC. The advantages of the quantum algorithm are more obvious in the plot from certain

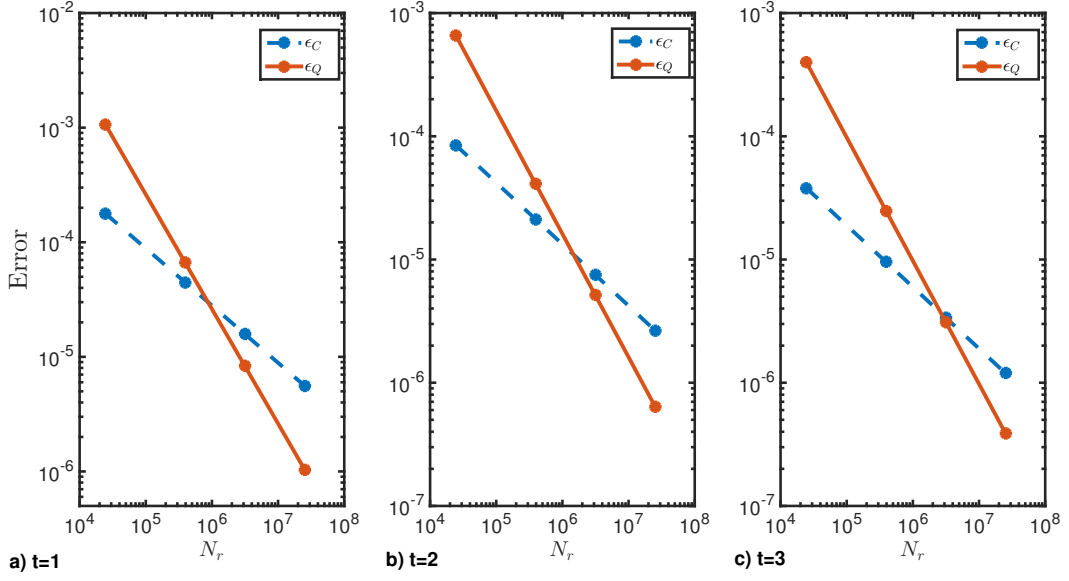


**Figure 3.** Estimates of the rate of reactant conversion obtained from classical simulations of the quantum-algorithm that would solve a reacting flow process using Curl’s model. The simulation parameters are the same as those of Fig. 1 except for different  $Da$  values. (a) Growth of the estimated rate of reactant conversion as a function of time in different cases of  $Da$  values. For  $Da=0.1$ , the purple dotted line is the quantum-algorithm simulation for  $N_r = 2^{17} \times 24$  and the blue circles are for the accurate MC estimate  $\tilde{Z}_{MC}(t)$  obtained using  $N_r = 2^{20} \times 60$ . For  $Da=1$ , the green dotted line is the quantum-algorithm simulation for  $N_r = 2^{17} \times 24$  and the orange circles are for the accurate MC estimate  $\tilde{Z}_{MC}(t)$  obtained using  $N_r = 2^{20} \times 60$ . For  $Da \rightarrow \infty$ , the cyan dotted line is the quantum-algorithm simulation for  $N_r = 2^{17} \times 24$  and the yellow circles are for the accurate MC estimate  $\tilde{Z}_{MC}(t)$  obtained using  $N_r = 2^{20} \times 60$ . (b) Comparison of two estimated rates of reactant conversion for  $N_r = 2^{17} \times 24$  (black line) and  $N_r = 2^{20} \times 24$  (red line), given by the quantum algorithm, with  $\tilde{Z}_{MC}(t)$  and  $Da=1$ . A different scale than that of Fig.1(b) is used here because of the different error ranges in the two calculations. Smaller error values within these estimates are already obtained for values of  $N_r$  where  $m = 17$  and 20.

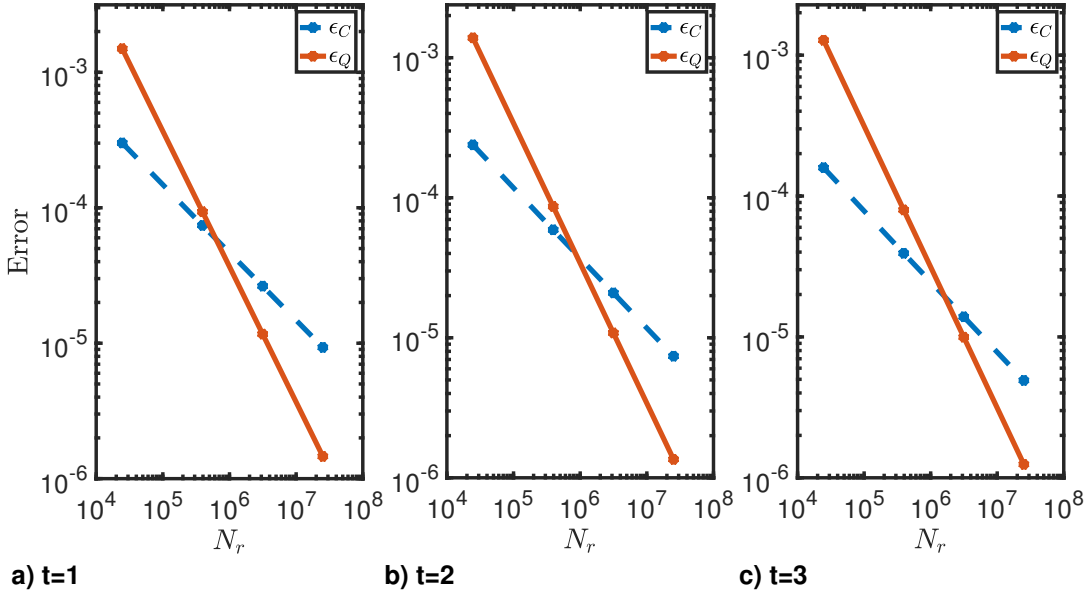
values of  $N_r$  for which  $\epsilon_Q \leq \epsilon_C$ . However, the genuine advantage is in the different scaling of the algorithm. Naturally, the resources required for a computation will always depend on prefactors that are dependent on the hardware, and classical and quantum algorithms will naturally run on different hardware.

#### 4. Concluding Remarks

Following on from Ref. [23], a quantum algorithm that leads to a quadratic speedup over classical MC methods in estimating statistical properties of a turbulent reacting flow has been demonstrated. In this demonstration, several restricting assumptions are imposed, and need to be discussed. First, turbulent mixing is simulated via the family of coalescence/dispersion (C/D) mixing model. The quantum algorithm as developed here would yield the same speedup in conjunction with MC simula-



**Figure 4.** Comparisons of errors output by the classical MC method ( $\epsilon_C$ ) and the simulated quantum algorithm ( $\epsilon_Q$ ) in the estimation of the rate of reactant conversion  $\mathcal{Z}_{MC}(t)$  for the case  $\text{Da} \rightarrow \infty$ . The results are for values of  $t = 1, 2, 3$  and different  $N_r = 2^m \times L$ , with  $m = 10, 14, 17, 20$  and  $L = 24$ .



**Figure 5.** Comparisons of errors output by the classical MC method ( $\epsilon_C$ ) and the simulated quantum algorithm ( $\epsilon_Q$ ) in the estimation of the rate of reactant conversion  $\mathcal{Z}_{MC}(t)$  for the case  $\text{Da} = 1$ . The results are for values of  $t = 1, 2, 3$  and different  $N_r = 2^m \times L$ , with  $m = 10, 14, 17, 20$  and  $L = 24$ .

tion of alternative mixing closures [9, 42–55]. Second, is the assumption of a single-step, irreversible binary reaction. Determination of the reactants' conversion rate with this model is very useful from

both theoretical [2, 36, 38, 47, 52, 55] and practical standpoints in analysis of plug flow reactors [4, 56–58]. Extensions to PDF calculations with reversible reactions and multi-step kinetics are straightforward. Third, with the assumption of homogeneity, the effects of advection and spatial diffusion are masked in the formulations. Extension to an inhomogeneous flow simulation is also straightforward. To do so, more modelling is obviously required to describe the PDF evolution. Finally, with the single-point PDF descriptor, no information is available pertaining to the evolution of the turbulence time/length scales. Therefore, in the context as considered, the mixing frequency is prescribed arbitrarily. In practical applications, this must be provided by external means (turbulence models, experimental data), or a multi-point PDF formulation [51]. In all of these cases, a quantum speed-up would be realized similar to that in the examples presented here.

As with Shor’s algorithm for factorization of integer numbers, quantum algorithms such as the one presented here will require large-scale quantum computing hardware in order to encode and solve real-world problems of interest in aerospace. Because such hardware is quite probably a decade or two away, it is vital to use this basis as a starting point to connect communities of computational fluid dynamics and developers of quantum algorithms. This work should form the basis for further investigation of quantum algorithms for reacting flows, bridging between these two fields. We hope that this will lead to further optimized algorithms (and perhaps optimized quantum hardware) for dealing with turbulent reacting flow problems, including algorithms for specific applications that could be usefully implemented on much nearer-term hardware.

## Acknowledgements

This work was supported by AFOSR Grant FA9550-12-1-0057, Quantum Speedup for Turbulent Combustion Simulations, which brought together the authors from physics, quantum information, and engineering. Results were obtained using the EPSRC-funded ARCHIEWeSt High Performance Computer ([www.archie-west.ac.uk](http://www.archie-west.ac.uk)). EPSRC Grant No. EP/K000586/1. Additional computational resources were provided by the Center for Research Computing (CRC.pitt.edu) at the University of Pittsburgh

## Nomenclature

Da. Damköhler number.

$F$ . fuel.  
 $\mathcal{J}$ . Shvab-Zeldovich variable.  
 $j$ . the ordinal number of ancillary qubits in PEA algorithm,  $1 \leq j \leq m$ .  
 $k$ . the ordinal number of repetitions,  $1 \leq k \leq N_r$ .  
 $L$ . the number of realizations of the quantum algorithm.  
 $l$ . the ordinal number of realizations of the quantum algorithm,  $1 \leq l \leq L$ .  
 $m$ . the number of ancillary qubits used in the quantum phase estimation algorithm.  
 $N_r$ . the number of repetitions in Monte-Carlo simulations.  
 $N_p$ . the number of particles in Monte-Carlo simulations.  
 $N_t$ . the number of time steps in Monte-Carlo simulations.  
 $O$ . oxidizer.  
 $P$ . PDF.  
Pr. Probability  
 $p_s$ . probability of success of each execution of quantum PEA algorithm.  
 $\mathcal{Q}$ . approximated PDF of MC method.  
 $q$ . the ordinal number of time steps in MC method,  $1 \leq q \leq N_t$ .  
 $r$ . the stoichiometric coefficient.  
 $t$ . physical time.  
 $t_q$ . physical time at  $q$ -th time step  
 $W$ . area weight of the reactant.  
 $Z$ . reactant conversion rate.

### ***Greek letters***

$\xi$ . the composition space for the mixture fraction.  
 $\Delta t$ . the time step of MC simulations.  
 $\Psi$ . the composition field.  
 $\psi$ , the marginal composition field.  
 $\sigma$ . the standard deviation of estimation via classical MC methods.  
 $\epsilon$ . the estimation error.  
 $\theta$ . the eigenphase.



## ***Subscripts***

0. time zero (inlet of plug flow reactor).

$C$ . classical algorithm.

st. stoichiometric.

$MC$ . Monte-Carlo simulations.

$Q$ . quantum algorithm.

## ***Other symbols***

$\langle \rangle$ . probability average.

$\hat{\cdot}$ . estimation value

$\tilde{\cdot}$ . a very accurate estimation value used as a normalization factor.

$|\cdot\rangle$ . ket notation of a quantum state.

$\langle \cdot|$ . bra notation of a quantum state.

## **References**

- [1] W.R. Hawthorne, D.S. Wedell, and H.C. Hottel, *Mixing and Combustion in Turbulent Gas Jets*, in *3rd Symp. on Combustion, Flames and Explosion Phenomena*, The Combustion Institute, Pittsburgh, PA, 1949, pp. 266–288.
- [2] H.L. Toor, *Mass Transfer in Dilute Turbulent and Nonturbulent Systems with Rapid Irreversible Reactions and Equal Diffusivities*, *AIChE J.* 8 (1962), pp. 70–78.
- [3] E.E. O’Brien, *Turbulent Mixing of Two Rapidly Reacting Chemical Species*, *Phys. Fluids* 14 (1971), pp. 1326–1331.
- [4] R.S. Brodkey (ed.), *Turbulence in Mixing Operation*, Academic Press, New York, NY, 1975.
- [5] S.H. Frankel, C.K. Madnia, and P. Givi, *Comparative Assessment of Closures for Turbulent Reacting Flows*, *AIChE J.* 39 (1993), pp. 899–903.
- [6] S.B. Pope, *PDF Methods for Turbulent Reactive Flows*, *Prog. Energ. Combust.* 11 (1985), pp. 119–192.
- [7] P. Givi, *Model Free Simulations of Turbulent Reactive Flows*, *Prog. Energ. Combust.* 15 (1989), pp. 1–107.
- [8] W. Kollmann, *The PDF Approach to Turbulent Flow*, *Theor. Comp. Fluid Dyn.* 1 (1990), pp. 249–285.
- [9] C. Dopazo, *Recent Developments in PDF Methods*, in *Turbulent Reacting Flows*, P.A. Libby and F.A. Williams, eds., chap. 7, Academic Press, London, England, 1994, pp. 375–474.

- [10] R.O. Fox, *Computational Models for Turbulent Reacting Flows*, Cambridge University Press, Cambridge, UK, 2003.
- [11] D.C. Haworth, *Progress in Probability Density Function Methods for Turbulent Reacting Flows*, Prog. Energ. Combust. 36 (2010), pp. 168–259.
- [12] D.C. Haworth and S.B. Pope, *Monte Carlo Solutions of a Joint PDF Equation for Turbulent Flows in General Orthogonal Coordinates*, J. Comput. Phys. 72 (1987), pp. 311–346.
- [13] E.A. Martinez, C.A. Muschik, P. Schindler, D. Nigg, A. Erhard, M. Heyl, P. Hauke, M. Dalmonte, T. Monz, P. Zoller, and R. Blatt, *Real-Time Dynamics of Lattice Gauge Theories with a Few-Qubit Quantum Computer*, Nature 534 (2016), pp. 516–519.
- [14] S. Debnath, N.M. Linke, C. Figgatt, K.A. Landsman, K. Wright, and C. Monroe, *Demonstration of a Small Programmable Quantum Computer with Atomic Qubits*, Nature 536 (2016), pp. 63–66.
- [15] T. Monz, D. Nigg, E.A. Martinez, M.F. Brandl, P. Schindler, R. Rines, S.X. Wang, I.L. Chuang, and R. Blatt, *Realization of a Scalable Shor Algorithm*, Science 351 (2016), pp. 1068–1070.
- [16] J. Kelly, R. Barends, A.G. Fowler, A. Megrant, E. Jeffrey, T.C. White, D. Sank, J.Y. Mutus, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, I.C. Hoi, C. Neill, P.J.J. O’Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, A.N. Cleland, and J.M. Martinis, *State Preservation by Repetitive Error Detection in a Superconducting Quantum Circuit*, Nature 519 (2015), pp. 66–69.
- [17] M.H. Devoret and R.J. Schoelkopf, *Superconducting Circuits for Quantum Information: An Outlook*, Science 339 (2013), pp. 1169–1174.
- [18] D.D. Awschalom, L.C. Bassett, A.S. Dzurak, E.L. Hu, and J.R. Petta, *Quantum Spintronics: Engineering and Manipulating Atom-Like Spins in Semiconductors*, Science 339 (2013), pp. 1174–1179.
- [19] J. Clarke and F.K. Wilhelm, *Superconducting Quantum Bits*, Nature 453 (2008), pp. 1031–1042.
- [20] R. Blatt and D. Wineland, *Entangled States of Trapped Atomic Ions*, Nature 453 (2008), pp. 1008–1015.
- [21] A. Montanaro, *Quantum Algorithms: An Overview*, Npj Quantum Information 2 (2016), pp. 15023 EP –.
- [22] P.W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM J. Comp. 26(5) (1997), pp. 1484–1509.
- [23] G. Xu, A.J. Daley, P. Givi, and R.D. Somma, *Turbulent Mixing Simulation via a Quantum Algorithm*, AIAA Journal 56 (2018), pp. 687–699.
- [24] L.K. Grover, *A Fast Quantum Mechanical Algorithm for Database Search*, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (1996), pp. 212–219.
- [25] A. Chowdhury and R.D. Somma, *Quantum Algorithms for Gibbs Sampling and Hitting-Time Estimation*, Quant. Inf. Comp. 17 (2017), pp. 0041–0064.
- [26] A. Montanaro, *Quantum Speedup of Monte Carlo Methods*, Proc. Roy. Soc. Ser. A 471 (2015), p. 20150301.
- [27] M. Szegedy, *Quantum Speed-Up of Markov Chain Based Algorithms*, Proceedings of the 45th Annual

- IEEE Symposium on Foundations of Computer Science (2004), pp. 32–41.
- [28] E. Knill, G. Ortiz, and R. Somma, *Optimal Quantum Measurements of Expectation Values of Observables*, Phys. Rev. A 75 (2007), p. 012328.
  - [29] R.D. Somma, S. Boixo, H. Barnum, and E. Knill, *Quantum Simulations of Classical Annealing Processes*, Phys. Rev. Lett. 101 (2008), pp. 130504–4.
  - [30] F. Magniez, A. Nayak, J. Roland, and M. Santha, *Search via Quantum Walk*, Proceedings of the 39th Symposium on the Theory of Computing (2006), pp. 575–584.
  - [31] V. Giovannetti, S. Lloyd, and L. Maccone, *Quantum Metrology*, Phys. Rev. Lett. 96 (2006), p. 010401.
  - [32] D. Poulin and P. Wocjan, *Preparing Ground States of Quantum Many-Body Systems on a Quantum Computer*, Phys. Rev. Lett. 102 (2009), p. 130503.
  - [33] V. Giovannetti, S. Lloyd, and L. Maccone, *Quantum-Enhanced Measurements: Beating the Standard Quantum Limit*, Science 306 (2004), p. 1330.
  - [34] J. Janicka, W. Kolbe, and W. Kollmann, *Closure of the Transport Equation for the Probability Density Function of Turbulent Scalar Field*, J. Non-Equil. Thermodyn. 4 (1979), pp. 47–66.
  - [35] S.B. Pope, *An Improved Turbulent Mixing Model*, Combust. Sci. Technol. 28 (1982), pp. 131–145.
  - [36] G. Kosály and P. Givi, *Modeling of Turbulent Molecular Mixing*, Combust. Flame 70 (1987), pp. 101–118.
  - [37] R.L. Curl, *Dispersed Phase Mixing: I. Theory and Effects in Simple reactors*, AIChE J. 9 (1963), pp. 175–181.
  - [38] R.W. Bilger, *Turbulent Flows with Nonpremixed Reactants*, in *Turbulent Reacting Flows*, P.A. Libby and F.A. Williams, eds., chap. 3, Springer-Verlag, Heidelberg, 1980, pp. 65–113.
  - [39] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.
  - [40] P. Kaye, R. Laflamme, and M. Mosca, *An Introduction to Quantum Computing*, Oxford University Press, USA, 2007.
  - [41] D. Nagaj, P. Wocjan, and Y. Zhang, *Fast Amplification of QMA*, Quantum Info. Comput. 9 (2009), pp. 1053–1068.
  - [42] Z. Ren and S.B. Pope, *An Investigation of the Performance of Turbulent Mixing Models*, Combust. Flame 136 (2004), pp. 208–216.
  - [43] L. Valiño and C. Dopazo, *A Binomial Langevin Model for Turbulent Mixing*, Phys. Fluids A 3 (1991), pp. 3034–3037.
  - [44] S.B. Pope, *A Model for Turbulent Mixing Based on Shadow-Position Conditioning*, Phys. Fluids 25 (2013), p. 110803.
  - [45] S. Subramaniam and S.B. Pope, *Comparison of Mixing Model Performance for Nonpremixed Turbulent Reactive Flow*, Combust. Flame 117 (1999), pp. 732–754.
  - [46] F.A. Jaber, R.S. Miller, C.K. Madnia, and P. Givi, *Non-Gaussian Scalar Statistics in Homogeneous*

- Turbulence*, J. Fluid Mech. 313 (1996), pp. 241–282.
- [47] S.H. Frankel, P.A. McMurtry, and P. Givi, *Linear Eddy Modeling of Reactant Conversion and Selectivity in Homogeneous Turbulence*, AIChE J. 41 (1995), pp. 258–266.
  - [48] S.B. Pope, *Mapping Closures for Turbulent Mixing and Reaction*, Theor. Comp. Fluid Dyn. 2 (1991), pp. 255–270.
  - [49] R.S. Miller, S.H. Frankel, C.K. Madnia, and P. Givi, *Johnson-Edgeworth Translation for Probability Modeling of Binary Scalar Mixing in Turbulent Flows*, Combust. Sci. Technol. 91 (1993), pp. 21–52.
  - [50] B. Merci, D. Roekaerts, B. Naud, and S.B. Pope, *Comparative Study of Micromixing Models in Transported Scalar PDF Simulations of Turbulent Nonpremixed Bluff Body Flames*, Combust. Flame 146 (2006), pp. 109–130.
  - [51] T.L. Jiang, F. Gao, and P. Givi, *Binary and Ternary Scalar Mixing by Fickian Diffusion-Some Mapping Closure Results*, Phys. Fluids A 4 (1992), pp. 1028–1035.
  - [52] S.H. Frankel, T.L. Jiang, and P. Givi, *Modeling of Isotropic Reacting Turbulence by a Hybrid Mapping-EDQNM Closure*, AIChE J. 38 (1992), pp. 535–543.
  - [53] F.A. Jaber and P. Givi, *Inter-Layer Diffusion Model of Scalar Mixing in Homogeneous Turbulence*, Combust. Sci. Technol. 104 (1995), pp. 249–272.
  - [54] A.Y. Klimenko and S.B. Pope, *The Modeling of Turbulent Reactive Flows Based on Multiple Mapping Conditioning*, Phys. Fluids 15 (2003), pp. 1907–1925.
  - [55] C.K. Madnia, S.H. Frankel, and P. Givi, *Reactant Conversion in Homogeneous Turbulence: Mathematical Modeling, Computational Validations and Practical Applications*, Theor. Comp. Fluid Dyn. 4 (1992), pp. 79–93.
  - [56] H.L. Toor, *The Non-Premixed Reaction:  $A + B \rightarrow \text{Products}$* , in Brodkey [4], pp. 123–166.
  - [57] R.S. Brodkey, *Fundamental of Turbulent Motion*, Chem. Eng. Comm. 8 (1981), pp. 1–23.
  - [58] A. Dutta and J.M. Tarbell, *Closure Models for Turbulent Reacting Flows*, AIChE J. 35 (1989), pp. 2013–2027.